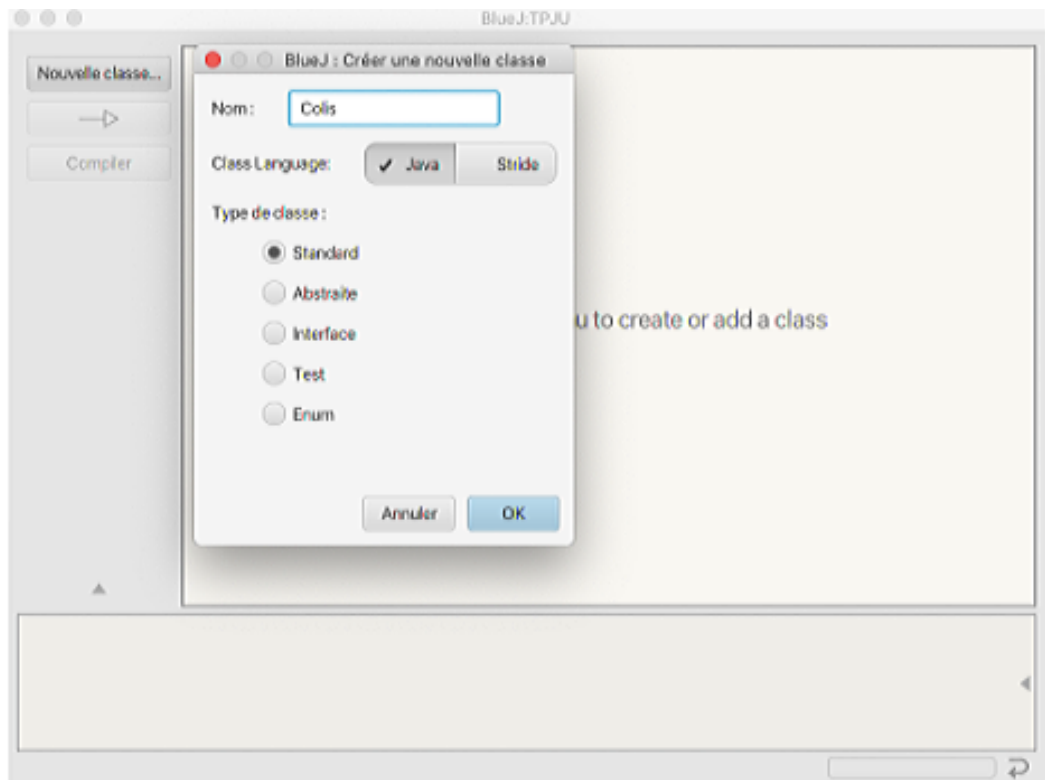


MARS 2020

TUTORIAL CONCEPT ORIENTE OBJET

Julienne ISHA &
Marie VOUILLOT





On commence par créer notre classe fétiche qui "fabriquera" des colis

```
public class Colis
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre

    private double poids;
    private String adresse;
    private String destinataire;

    /**
     * Constructeur d'objets de classe Colis
     */
    public Colis(String adresse, String destinataire)
    {
        // initialisation des variables
        this.adresse=adresse;
        poids=0.1;
        this.destinataire=destinataire;

    }

    /**
     * @un getter pour le poids
     */
    public double getPoids()
    {
        return poids;
    }

    /**
     * @un getter pour l'adresse
     */
    public String getAdresse()
    {
        return adresse;
    }

    /**
     * @un getter pour le destinataire
     */
    public String getDestinataire()
    {
        return destinataire;
    }

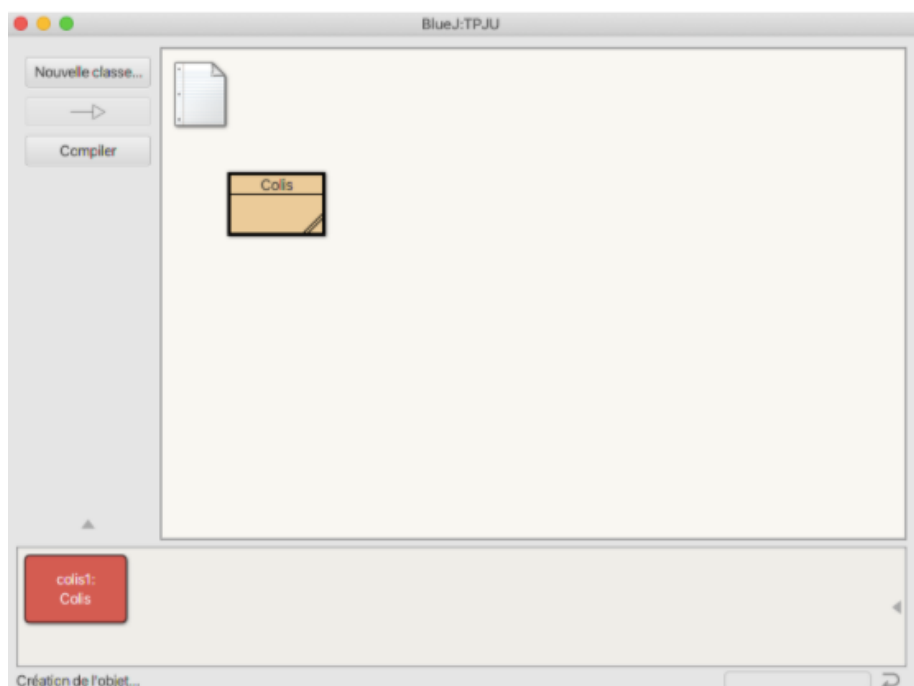
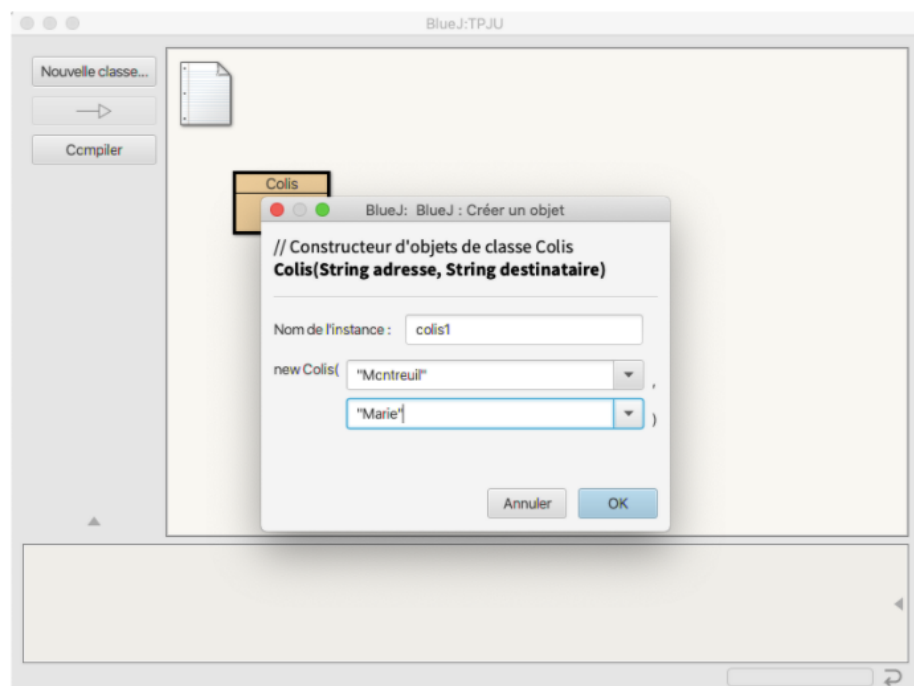
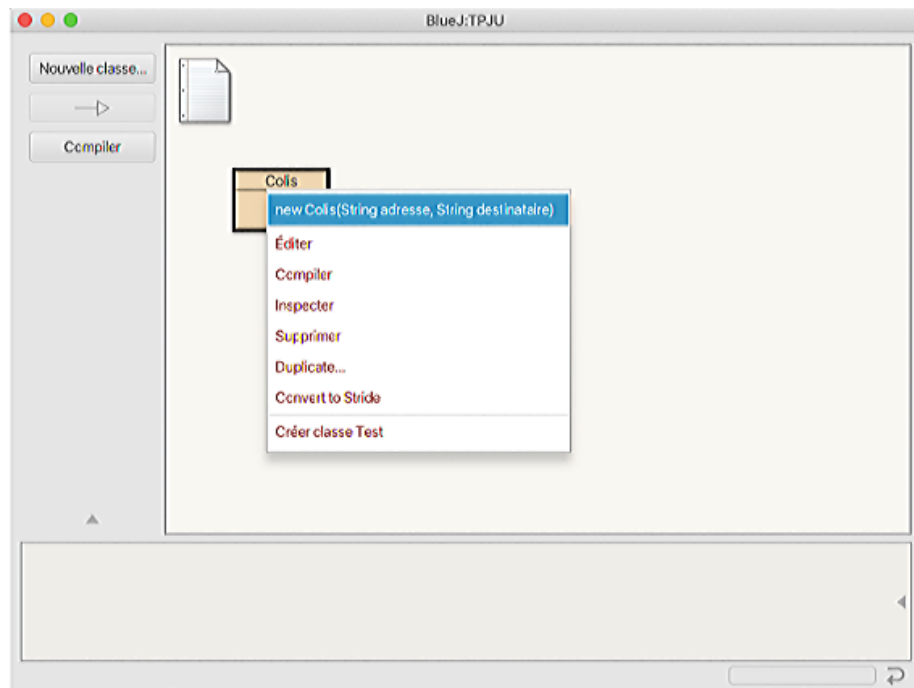
    /**
     *Méthode pour ajouter de la charge dans le colis
     */
    public double chargementColis( double charge)
    {
        poids+=charge;
        return poids;
    }
}
```

Les colis auront chacun un poids, une adresse de livraison ainsi qu'un destinataire.

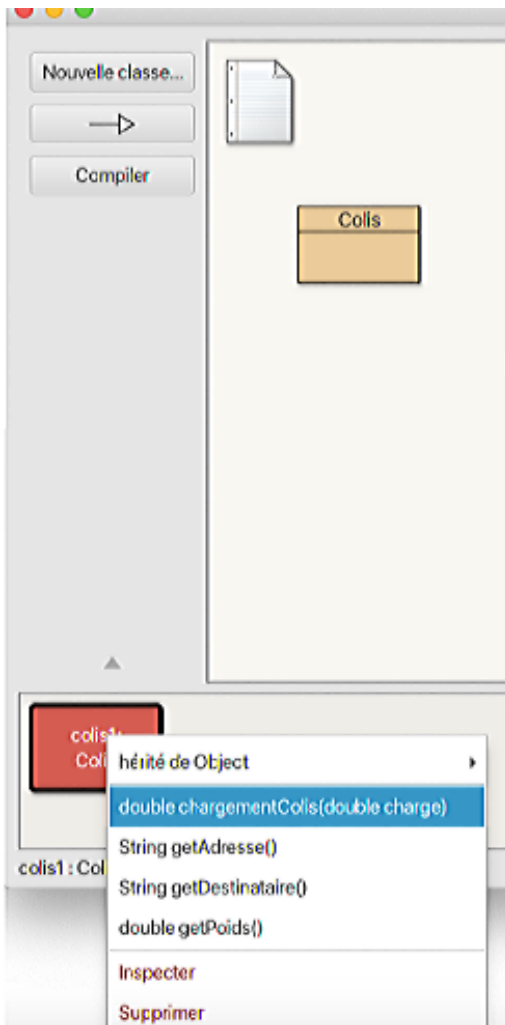
Le constructeur permet de "fabriquer" les colis. Ici on voit que l'adresse et le destinataire seront saisis par l'utilisateur mais le poids sera forcément de 0,100g lors de la création.

il y a également des méthodes pour récupérer les valeurs des variables.

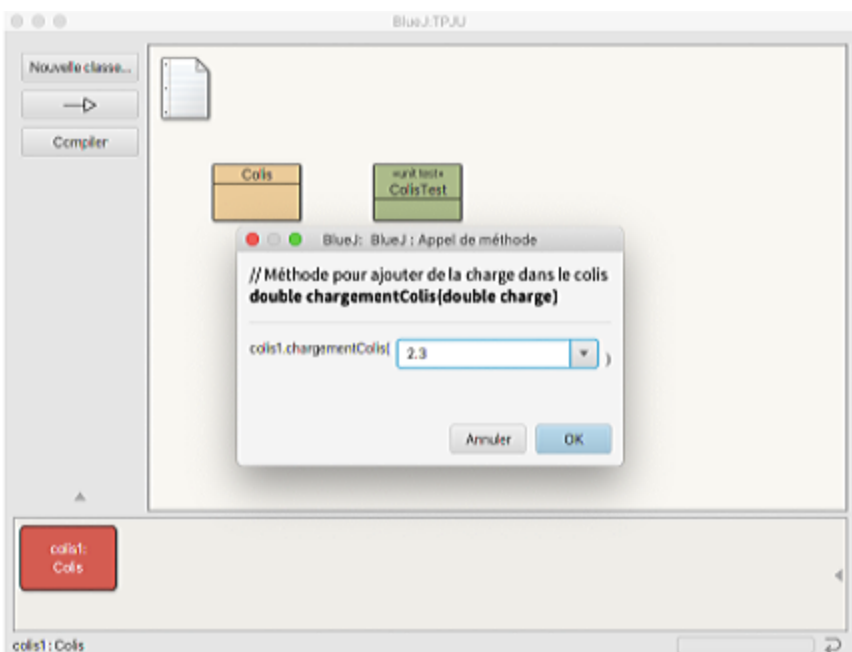
Enfin on a une méthode chargementColis, qui prend en attribut une charge et qui l'ajoute au poids du colis. Cette méthode représente l'action de remplir un colis.



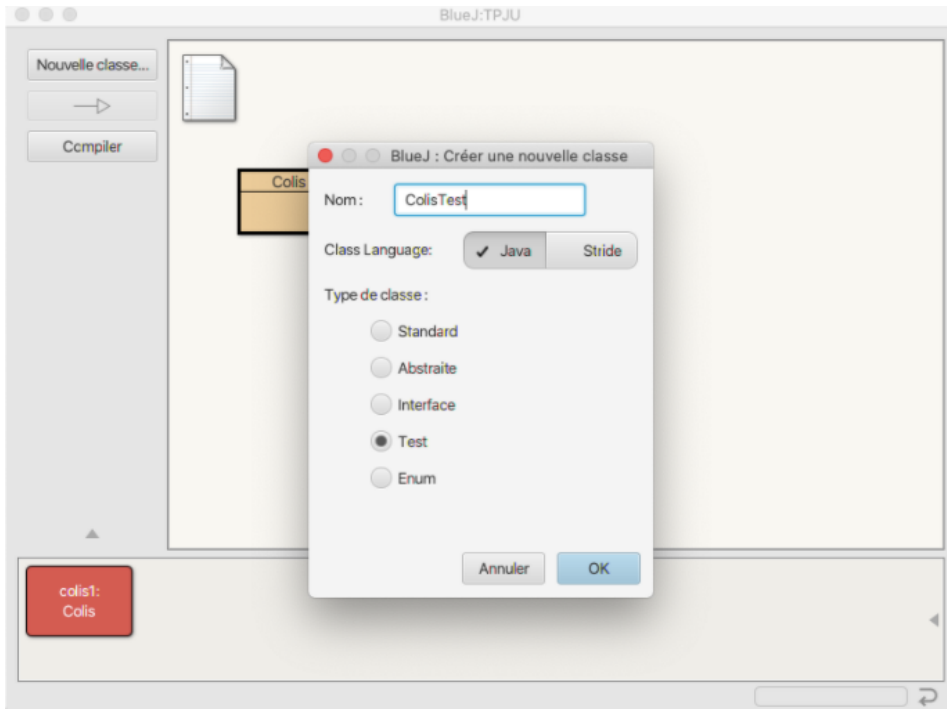
ici, on crée une instance de notre classe Colis.



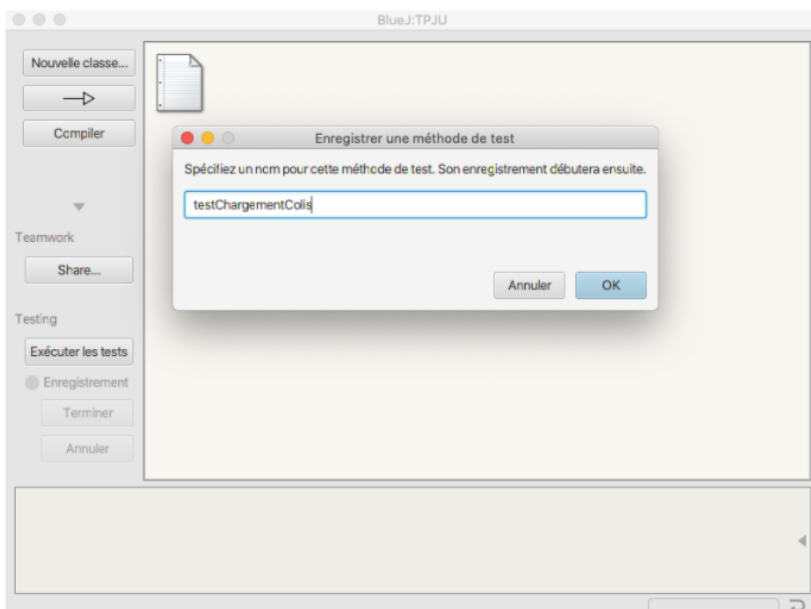
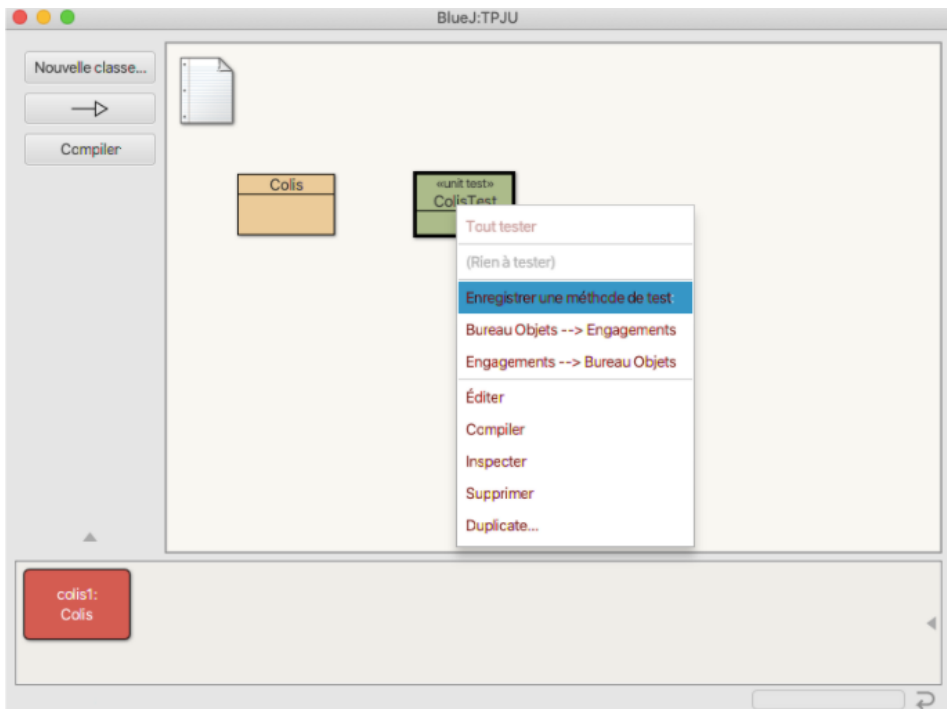
On ajoute de la charge dans le colis avec la méthode **chargementColis()**

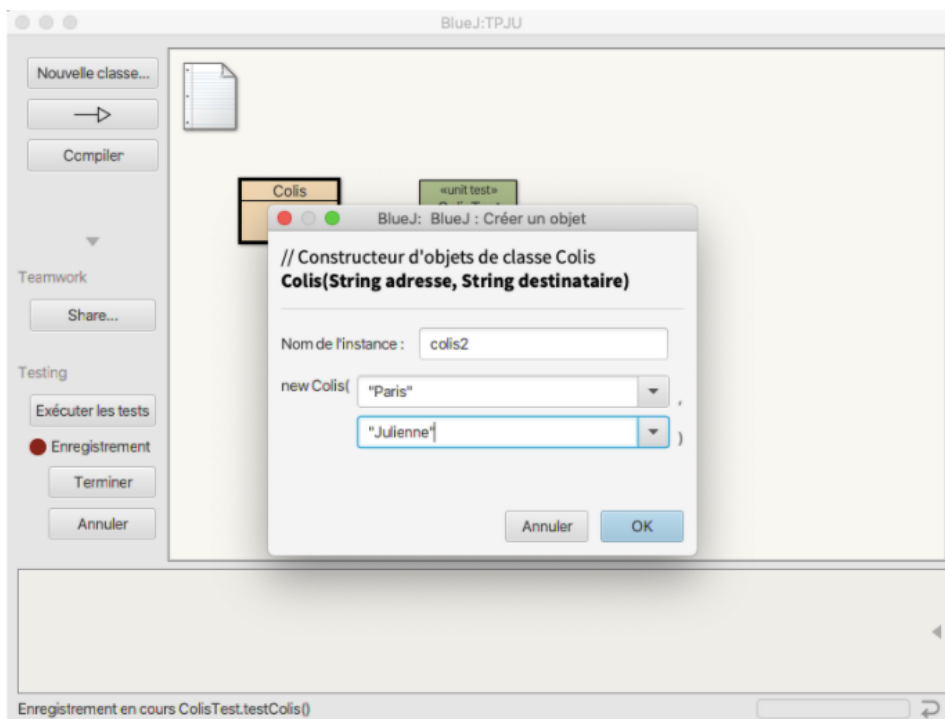


Ici par exemple on ajoute 2,3 kg dans notre colis.

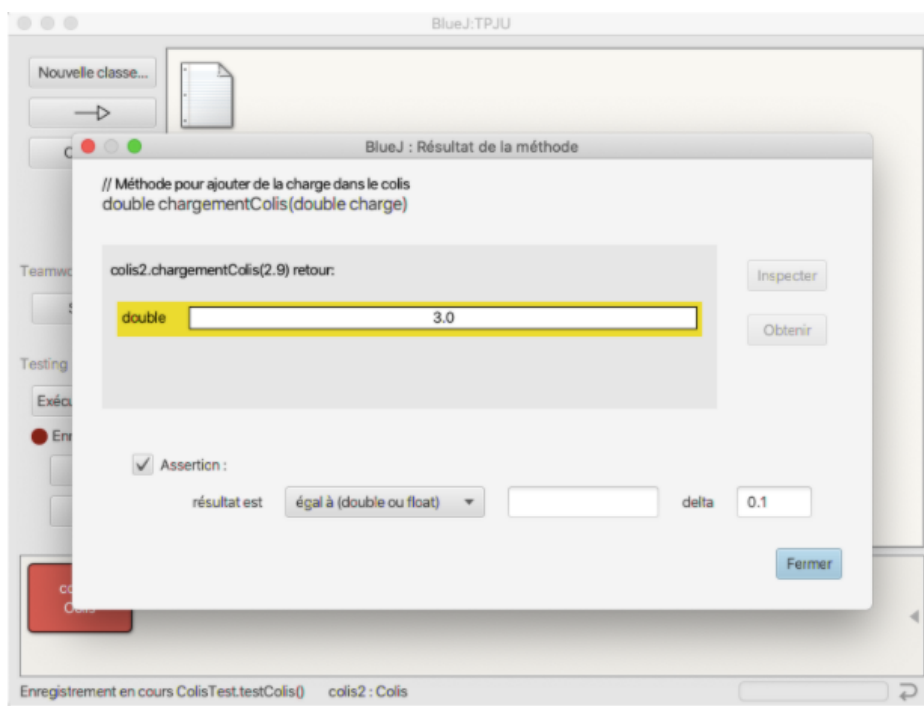
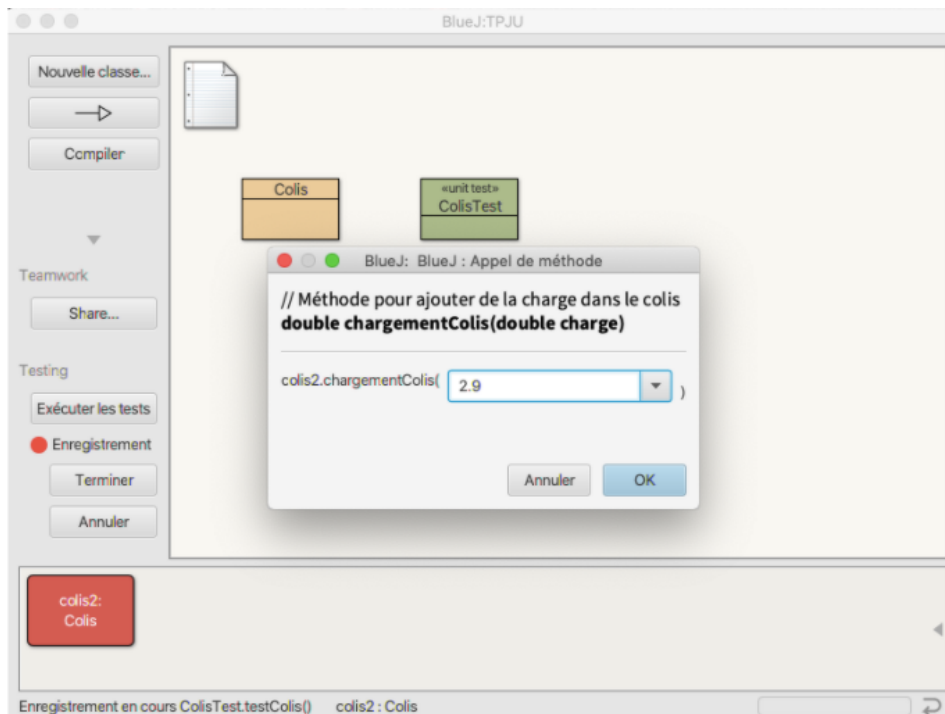


On crée une classe de test pour vérifier que notre méthode de chargement renvoie le résultat que l'on attend.

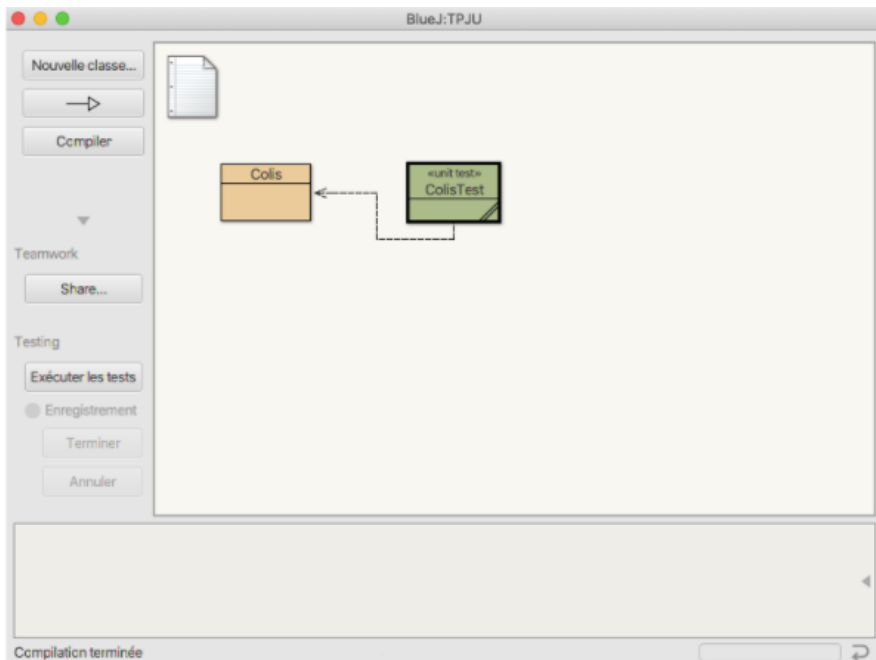




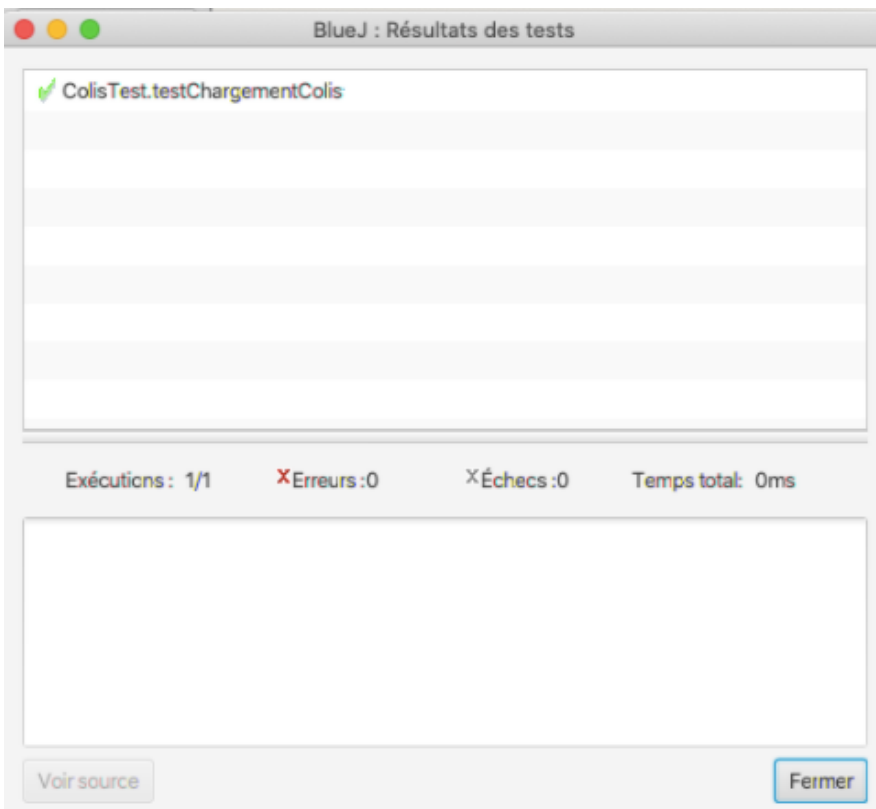
Pour cela on va refaire le processus de création d'un colis et de son chargement et le test sera enregistré tout le long.



Ici il faut entrer le poids que l'on est censé trouver suite à l'ajout d'une charge



Une fois
l'enregistrement
du test terminé,
on peut exécuter
le test
On peut voir que
le test a
fonctionné. La
fonction
ChargementColis
fonctionne donc
bien.



```

/**
public class bonLivraison
{
    // variables d'instance - remplacez l'exemple qui suit par le vôtre
    private double prix;
    public Colis c;

    /**
     * Constructeur d'objets de classe bonLivraison
     */
    public bonLivraison()
    {
        // initialisation des variables d'instance
        prix = 0;
    }

    public void setColis(Colis c){
        this.c=c;
    }
    public Colis getColis() {
        return this.c;
    }

    /**
     * @un getter pour le prix
     */
    public double getPrix()
    {
        return prix;
    }

    /**
     * @un getter pour le poids
     */
    public double getPoids()
    {
        double poids=this.c.getPoids();
        return poids;
    }
}

```

```

/**
 * Un exemple de méthode - remplacez ce commentaire par le vôtre
 *
 * @param y    le paramètre de la méthode
 * @return     la somme de x et de y
 */
public double calculPrix()
{
    prix=c.getPoids()*10.5;
    return prix;
}
}

```

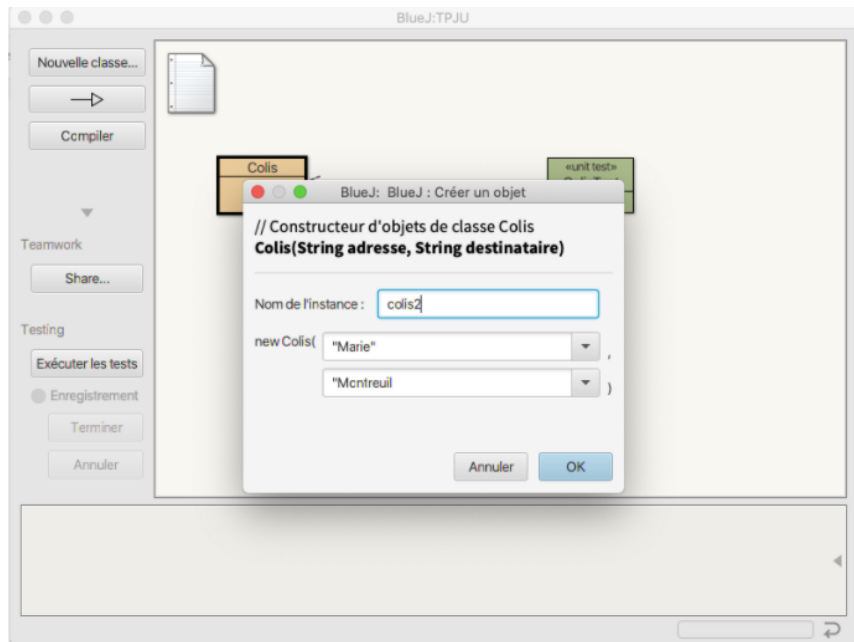
asse compilée - pas d'erreur de syntaxe

enregistré

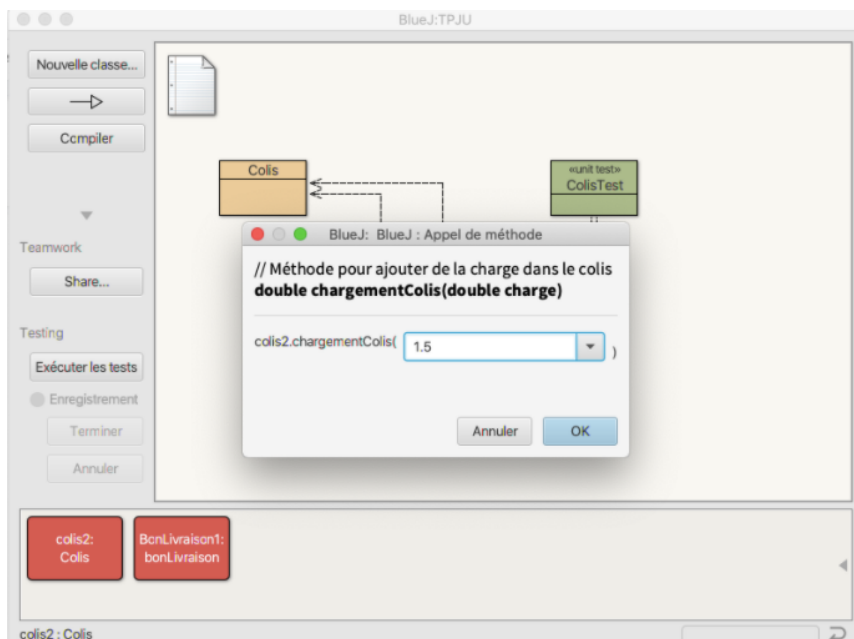
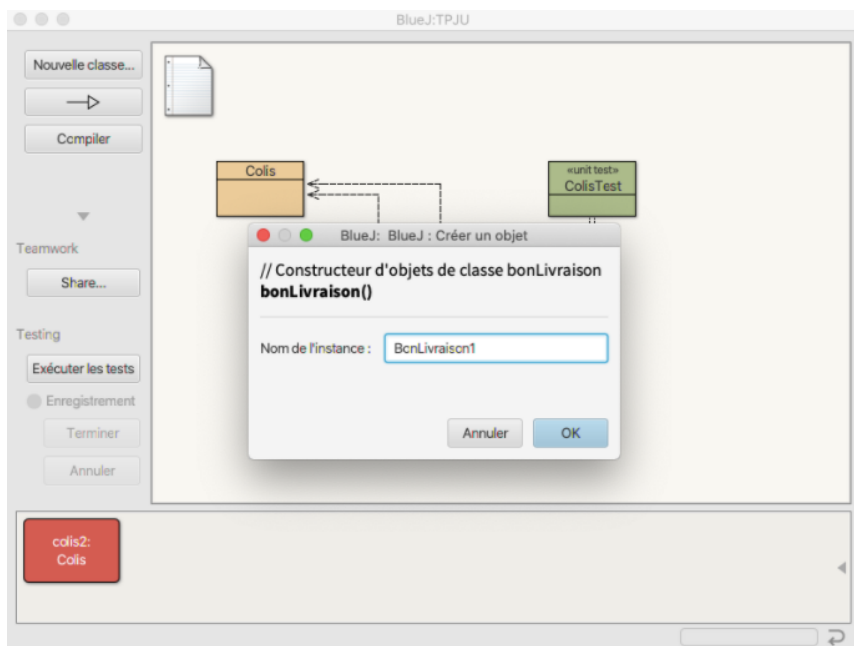
On crée maintenant une classe "bonLivraison". Sur le bon de livraison on aura le prix de la livraison, et il sera "lié" à un seul colis.

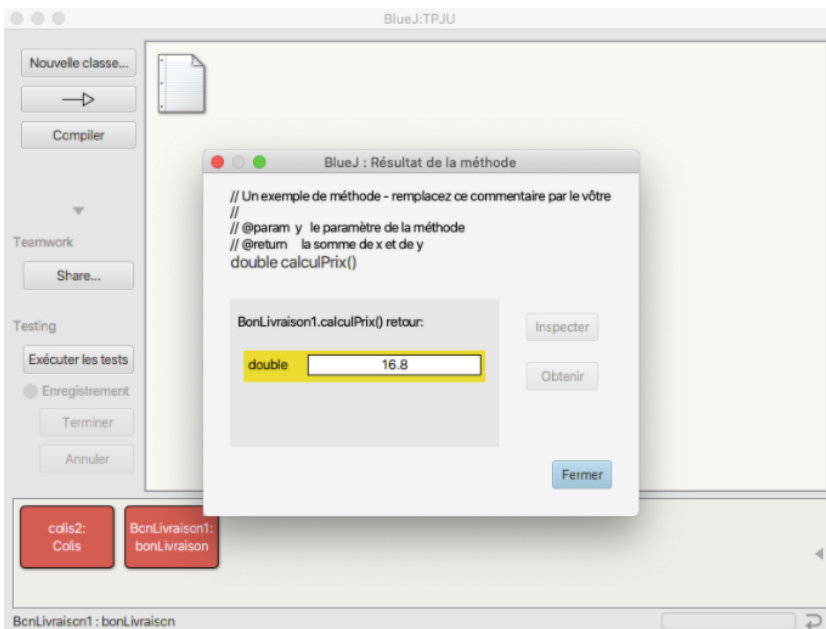
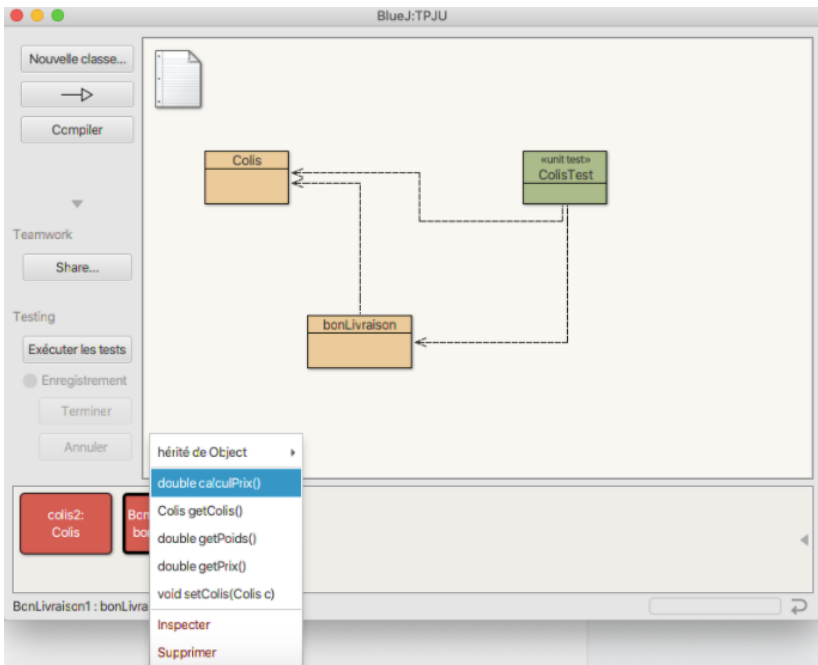
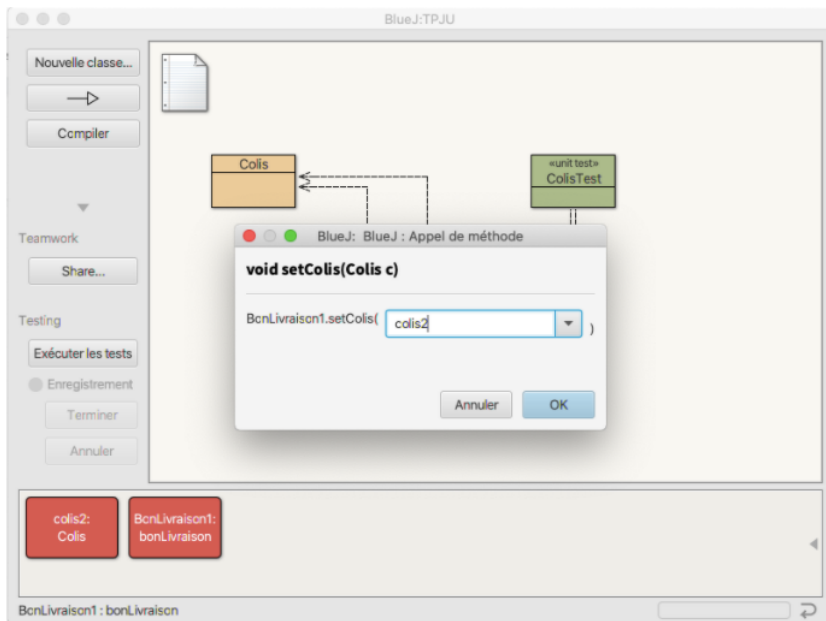
La méthode setColis permet d'attribuer un colis à un bon de livraison et la méthode getPoids() permettra de récupérer son poids par exemple.

Enfin la méthode calculPrix() permettra de calculer le prix de la livraison en fonction du poids du colis. Ici, 1kg coûtera 10,50€.

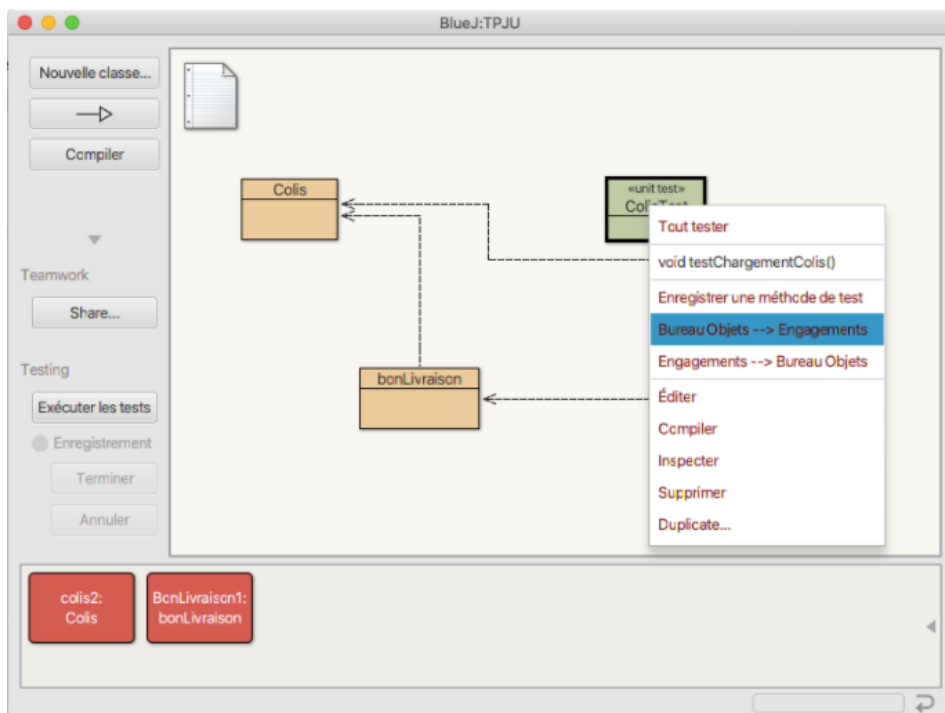


Ici, on va créer un colis, le remplir, puis créer le bon de livraison correspondant (avec le prix de livraison calculé grâce a la méthode `calculPrix`).





Par exemple ici le prix sera de 16,80€ pour un colis de 1,6kg.



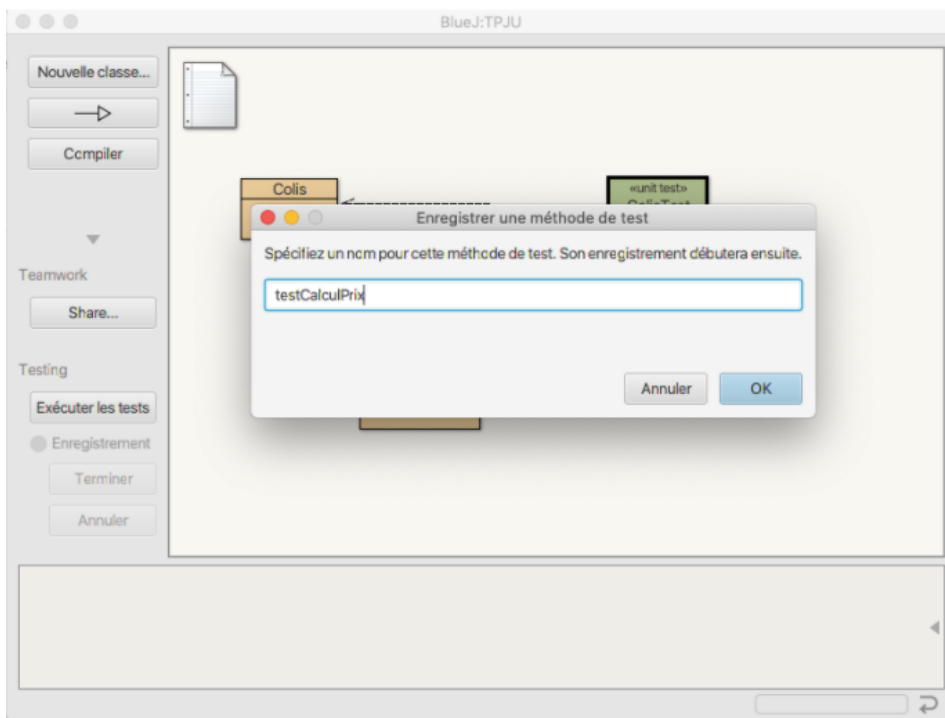
On ajoute ces objets à la fixture de la classe de test (dans la méthode setUp() de la classe testColis.

Pourquoi faire ça ?

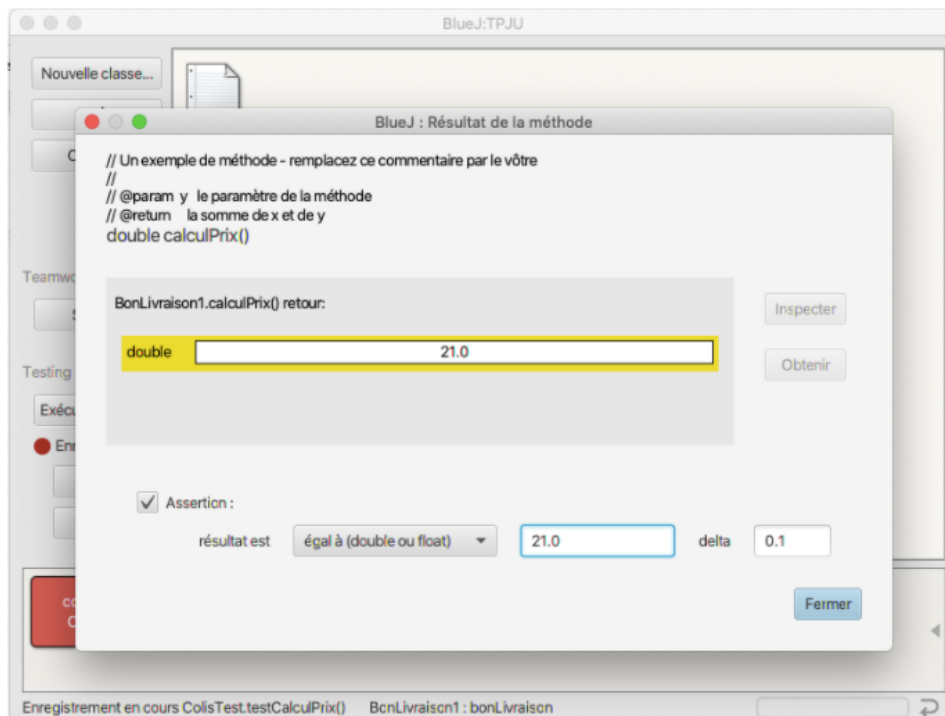
Lorsqu'on lancera un test, ces objets seront automatiquement créés afin que le test puisse se dérouler.

```
/**
 * Constructeur de la classe-test ColisTest
 */
public ColisTest()
{
}

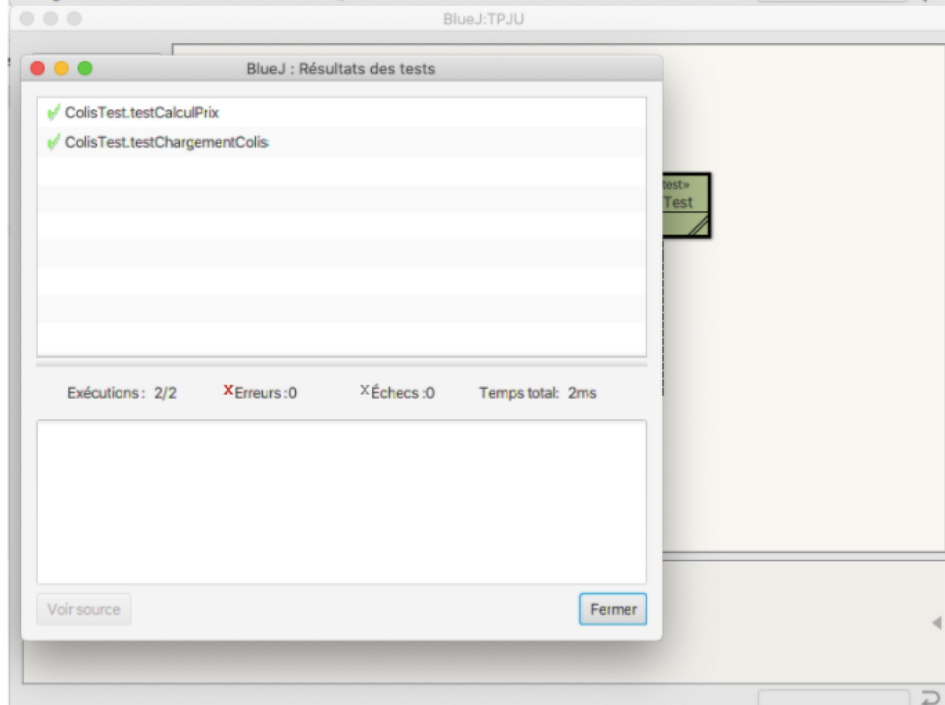
/**
 * Met en place les engagements.
 *
 * Méthode appelée avant chaque appel de méthode de test.
 */
@Before
public void setUp() // throws java.lang.Exception
{
    colis2 = new Colis("Marie", "Montreuil");
    BonLivraison1 = new bonLivraison();
    colis2.chargementColis(1.5);
    BonLivraison1.setColis(colis2);
    BonLivraison1.getPoids();
    BonLivraison1.getPrix();
    BonLivraison1.calculPrix();
}
```



On finit en ajoutant un test qui vérifie que le calcul du prix de livraison s'effectue bien.



Le prix pour un colis de 2kg doit être de 21€ (soit 2*10,5).



Une fois le test enregistré, on peut exécuter tous les tests. Ici tous les tests sont validés.